

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of

BRAMHILL et al.

Serial No.: 09/091,735

Filed: June 24, 1998

For: COPY PROTECTION OF DATA



Atty. Ref.: 36-1230

Group Art Unit: 2165

Examiner: Nguyen, C.

#14 Brief & Set 2me
19/12/02

August 28, 2002

APPEAL BRIEF

Assistant Commissioner for Patents
Washington, DC 20231

Sir:

Appellant hereby appeals the Rejection of January 29, 2002. An
Amendment/Response is being filed concurrently herewith.

REAL PARTY IN INTEREST

The real party in interest is the assignee, British Telecommunications public
limited company, a corporation of Great Britain.

RELATED APPEALS AND INTERFERENCES

The Appellant, the undersigned, and the assignee are not aware of any related
appeals or interferences which will directly affect or be directly affected by or have a
bearing on the Board's decision in this appeal.

08/30/2002 AWONDAF1 00000078 09091735

01 FC:120
02 FC:121

320.00 DP
200.00 BP

STATUS OF THE CLAIMS

Claims 1-8, 12, 14-18, 21-26 and 28-33 remain pending in this application.

Claims 1-8, 12, 14-18, 21-26 and 28-31 stand rejected by the Examiner, the rejections of which are appealed.

Claims 32 and 33 were not rejected in the January 29, 2002 Office Action. Appellant therefore presumes that these claims are allowable. For the Board's convenience, claims 32 and 33 are presented in the Appendix to this brief.

STATUS OF ANY AMENDMENT FILED SUBSEQUENT TO FINAL REJECTION

An Amendment is being concurrently filed herewith. By that Amendment, a typographical error in claim 31 is corrected (i.e., a semicolon is being deleted). While this amendment to claim 31 has not yet been formally entered by the Examiner, the claims as presented in the Appendix to this brief includes this change in anticipation that the Examiner will enter this amendment.

The claims as presented in the Appendix to this brief are as amended by the Amendments filed on November 16, 2001, April 24, 2001, June 20, 2000 and July 24, 1998.

CONCISE EXPLANATION OF THE INVENTION

The present invention relates to copy protecting data transmitted from a server to a client. An exemplary embodiment of the present invention is described below.

A webpage 7 containing copyright protected image data is downloaded from server 1 to client computer 3. In particular, client 3 uploads a request to server 1 for details of a webpage. The request to server 1 typically comprises a conventional hypertext file transfer protocol (HTTP) page request. Server 1 obtains or constructs the webpage and downloads the HTML code corresponding to the webpage to client computer 3. The downloaded HTML code may include a Java applet A1. Applet A1 is executed on client computer 3 using a Java interpreter within the webpage browser in order to prepare the browser to receive data to be displayed in, for example, a copyright protected region 12 of webpage 7.

The execution of applet A1 causes a request for a file of copyright protected data to be uploaded to server 1. Server 1 then performs an authentication step in order to determine whether it is safe to download the copyright protected data file requested by client computer 3. Assuming that client computer 3 passes this authentication step, server 1 prepares and transmits the copyrighted protected data file to client computer 3.

Preparing the copyright protected data file for downloading to client computer 3 involves watermarking the data to be downloaded. Watermarking gives additional security in the event the protected data is copied because knowledge of the source of copying can be determined from the watermark. (See step 10.2.) The watermarked data is hashed at server 1 using a copy of a hashing algorithm HA that was downloaded in applet A1 and a file specific session hashing key K_{SH} . This hashing ensures that sections of data are not removed and replaced by data such as to ensure

that for example a command “pay \$100” is not changed to “pay \$1.00.” (See step S10.3.) The data is then encrypted at server 1 using a copy of an encryption algorithm EA and a key K_E which are downloaded previously to client computer 3 in the Java byte codes of applet A1. The encryption algorithm EA forms a pair of algorithms, one of which is to encrypt and the other is to decrypt. (See step S10.4.) The resulting file is then wrapped in a proprietary file format which includes additional cryptographic protection techniques. (See step S10.5.) The resulting file of copyright protected data is then downloaded to client computer 3. (See step S11.)

The downloaded data file is then processed using applet A1 which was previously downloaded to client computer 3. Applet A1 allows the downloaded data file to be decrypted and checked for integrity (i.e., hash verified). Specifically, if the integrity of the content of a header is satisfactorily verified (see step S12.1), applet A1 checks whether it knows how to process files of the type specified in the header (see step S12.2). If the result of the check is satisfactory, applet A1 can make use of the specific copyright protected control information for the file present in header H when processing user requests for data manipulation. Specifically, the downloaded data file is decrypted using the encryption algorithm EA and the key K_E previously downloaded in applet A1. The integrity of the decrypted file is then verified. (See steps S12.6-12.7.) If this integrity check is satisfactory, applet A1 can display the content of the decrypted file in region 12 (See Fig. 4.) Accordingly, if all of the integrity checks are satisfied, the decrypted data may be copied or saved.

If the integrity check of the content header H is unsatisfactory, applet A1 determines that it does not know how to process files of the type specified in the header, or the integrity of the decrypted file is unsatisfactory, an error banner is displayed in region 12 (Fig. 4). (See step S12.3). If the downloaded copyright protected data file contains image data, the image is displayed, together with its imperceptible watermark in region 12. The user, however, cannot save or copy the image data. Because the Java enabled browser is executing an applet for the image data in region 12, the functions of a right mouse button including print, save or copy are disabled for region 12. Therefore, if a user clicks a right button of the mouse for region 12, no menu option is automatically provided for saving, copying or printing the displayed data in region 12 in order to prevent unauthorized copying.

When a user operating client computer 3 requests access to data, such as a particular webpage, instead of server 1 just sending the requested data to client computer 3, another program (an applet) is triggered to run at client computer 3 and take over the process for obtaining the requested data from server 1 and providing a controlled environment at client computer 3 in which the decrypted data may be viewed by the user. Since all of this executes in the background, the user would not normally be aware that anything else is going on beyond server 1 merely downloading the data as requested, unless the user tries to perform copyright-infringing actions at client computer 3.

CONCISE EXPLANATION OF THE ISSUES PRESENTED FOR REVIEW

Whether claims 1-8, 12, 14-18, 21-26 and 28-31 are made “obvious” under 35 U.S.C. §103 based on Yourdon in view of Dean et al (hereinafter “Dean”), Wobber et al (U.S. Patent No. 5,235,642, hereinafter “Wobber”), Richardson (PCT WO-A-9407204), and further in view of the IDS submitted on November 16, 2002 on “IMPRIMATUR” (hereinafter “IMPRIMATUR”).

WHETHER THE CLAIMS STAND OR FALL TOGETHER

Claims 1-8, 12, 14-18, 21 and 28-31 stand or fall together and do not stand or fall with any other claims.

Claims 22-26 stand or fall together and do not stand or fall with any other claims.

The specific reasons for each of the above groups of claim(s) standing or falling together or alone is provided below in the section entitled “Arguments with Respect to the Issue Presented for Review.”

**ARGUMENTS WITH RESPECT TO THE ISSUES PRESENTED FOR
REVIEW**

Claims 1-8, 12, 14-18, 21-26 and 28-31 are not “obvious” under 35 U.S.C. §103 over Yourdon in view of Dean, Wobber, Richardson and further in view of the IMPRIMATUR.

The January 29 Office Action states the following:

“Applicants’ arguments have been fully considered but they are not persuasive with previous cited references and further in view of submitted IDS

BRAMHILL et al.--Application No. 09/091,735

on "IMPRIMATUR" references for 35 U.S.C. §103(a) rejections. All the answers to the arguments on pp. 10-13 of the amendment received on 6/20/2000 are within court case decisions that the examiner submits as followings:..."

The Office Action then lists several court cases including quotations and/or caselaw citations. However, the Office Action fails to indicate any specific applications of these broad quotations and/or caselaw citations to the specific prior art and claimed invention at issue in the present application. Appellant thus respectfully requests clarification on how the quotations and caselaw citations specifically apply to the applied prior art and claimed invention.

The Office Action admits that Yourdon and Dean "do not directly address the specific problem of protecting from copying data, & authentication which have been downloaded from a server to a client, nor its solution as in claim 1." (See section 11 of the Office Action.) Appellant submits that Wobber, Richardson and IMPRIMATUR fail to remedy this deficiency of Yourdon and Dean. Indeed, Richardson is only discussed briefly in the Office Action with respect to dependent claims 5, 6 and 17. Even worse, there appears to be no mention of any details regarding how IMPRIMATUR relates to the claimed invention except its brief identification at the end of section 14 of the Office Action. Wobber is concerned with the problem of authenticating the source of requests for access to data in a distributed information system. Appellant submits that all of the security measures of Wobber to control access by a requestor to data is implemented at a server, not a program portion running on a client. Wobber further fails to teach or suggest selectively controlling

BRAMHILL et al.--Application No. 09/091,735

access to copy or save functions at the client in respect of unprotected data once access has been granted by the server.

Accordingly, Appellant respectfully submits that even if Yourdon, Dean and Wobber, Richardson and IMPRIMATUR were combined as proposed by the Office Action, the combination would not have taught or suggested all of the claimed limitations of independent claims 1, 28, 30 and 31. With respect to independent claim 22, the combination fails to teach or suggest, inter alia, “downloading data encrypted by means of the cryptographic key to the identified client, for decryption by the client using the key from the unique determinator.”

The Examiner completely fails to identify any feature in Yourdon, Dean, Wobber, Richardson or IMPRIMATUR corresponding to a program portion for running at a client to generate and upload a request for access to data and for rendering a cryptographically protected form of those data into an unprotected form. Moreover, the problems being addressed and the general context of these references are so different to that of the present invention that it is not possible to see how one skilled in the art might be motivated to change any one of these disclosures, and to add features absent from all those disclosures, to provide the claimed invention. Any suggestion to the contrary can only be based improperly on hindsight reasoning.

With respect to Yourdon, Yourdon discloses a conventional password protection technique in which a password is used to determine an end user's authorization to invoke certain functionality or to access certain data. Yourdon, however, does not explicitly state whether the password protection scheme disclosed

BRAMHILL et al.--Application No. 09/091,735

therein is implemented in respect of functionality or data residing at a server or that available at a client. If anything, since Yourdon states, “As long as I can control the security of my own server, then I can control the environment of the client platform, too (emphasis added),” immediately before the statement “The most likely scenario for the typical client-server application is one in which the application asks for a password, when appropriate, to determine the end-user’s authorization to invoke certain functionality or access certain data,” Appellant submits that Yourdon discloses the security of access to applets stored at a server. Yourdon fails to therefore discuss password protection for a client-server arrangement in the context of functionality and data residing at a client.

Yourdon mentions the need for Internet “applications” to be secure and hence for development tools to “help create secure access to functionality and data, as well as secure transmission of confidential data across the Internet.” However, “secure access” is a very general term and there are a great many aspects to “secure access.” Yourdon mentions a few of these aspects, but none is relevant to the type of security being provided in the present invention. For example, the Examiner cites Yourdon’s mention of “adding ‘digital signatures’ to an Internet-enabled Visual Basic applet, so that when you download it from the Web, you can be sure of its source and origin.” (See page 28, left column of Yourdon.) However, a “digital signature” does not restrict access to data, it is not a form of encryption, it is merely an indicator of origin. Digital signatures are not removed once they have been applied. There is no “decryption” to remove a digital signature. Digital signatures and other forms of

BRAMHILL et al.--Application No. 09/091,735

identification such as “digital watermarking” are thus not a form of cryptographic protection in that they do not in themselves protect or prevent access to data. In contrast, the present invention is concerned with what can be done with unprotected data at a client. Moreover, independent claims 1, 28, 30 and 31 require converting protected data to unprotected data at the client. Digital signatures are not applied to data so that they can be removed at the client. The digital signature remains embedded in data permanently so that the origin of the data may be determined in any future copy.

The Examiner also refers to Yourdon’s mention that “...the typical client-server application is one in which the application asks for a password, when appropriate, to determine the end user’s authorization to invoke certain functionality or access certain data. The application program will need to interact with the Web browser to encrypt/decrypt transmissions between the user’s workstation and the server,...” (See page 28, left column of Yourdon.) However, for the reasons discussed above, it would be clear to one skilled in the art that Yourdon is referring to authorization to access functionality and data at the server, not at the client. Password access to functionality and data is implemented at the server in a “typical” client-server application. The client merely obtains the password from a user and conveys it securely to the server.

Moreover, claim 1 requires “after the running of the program portion has begun and under control of the program portion at the client, converting the cryptographically protected data to an unprotected form and selectively controlling

BRAMHILL et al.--Application No. 09/091,735

access to copy or save functions at the client in respect of the data in its unprotected form (emphasis added).” That is, the present invention enables access to copy or save functions to be selectively controlled after running of the program portion has begun. In contrast, conventional password protection (such as that disclosed by Yourdon) prevents the running of an access-restricted program from even beginning (if, e.g., unauthorized password is provided beforehand). That is, an access-restricted program cannot be run if a password protection scheme is not first successfully passed (via an authorized password) under a conventional password protection scheme.

In the present invention, claim 1 is thus directed to features for restricting access to a data set after they arrive at a client. Independent claims 28, 30 and 31 require similar features. For example, claim 28 requires “...after the program portion is permitted to run at the source of the access request, in use:...to selectively control access to copy or save functions in respect of the data set when in said unprotected form (emphasis added).” Claim 30 requires “running a program at the client so that after running the program at the client has begun at the client, the program serves to both:...suppress client computer copy and save functions with respect to the unprotected copy of the requested data (emphasis added).” The authorization for a user at the client to access the downloaded data through a password is therefore not in question. That authorization would have typically have been established before downloading the data from the server to the client. If any password had been entered and conveyed to the server, that step would have been performed before any further operation of a program at the client is performed.

BRAMHILL et al.--Application No. 09/091,735

Moreover, the features of claims 1, 28, 30 and 31 are intended to restrict certain types of access by a user even if that user is “authorized” to, for example, reproduce a downloaded image on a screen. Entering a password is not necessarily going to alter the form of access that the user is able to have to the downloaded data when in an unprotected form at the client. Passwords (such as described by Yourdon) have an entirely different purpose in client-server applications to any that might be contemplated claims 1, 28, 30 and 31. For this reason, the Examiner’s reference to Yourdon’s belief that “It may be desirable to integrate packages like Pretty Good Privacy into our applications, so that end users can achieve higher levels of security with a public key password system” (see page 28, side column of Yourdon) is similarly irrelevant to these claims of the present invention.

As noted by the Office Action, Yourdon states that “Development tools must therefore help create secure access to functionality and data, as well as secure transmission of confidential data across the Internet.” (See page 28, left-side column of Yourdon.) This objective, however, is common to many security arrangements, and says nothing on how the “development tools” achieve this objective in the context of application programs or data.

Dean discusses security of Java in the context of the interplay between a Java applet and the underlying runtime system provided, for example, by a Java-enabled web browser. Dean is concerned with security weaknesses in the Java runtime system and browser architecture that might enable rogue applets to access resources that were

intended to be inaccessible, e.g., the file system, the CPU and network. Dean does not mention encryption.

In claim 1 of the present invention, a program portion running at the client is directed to “selectively controlling access to copy or save functions at the client in respect of the data in its unprotected form.” Independent claims 28, 30 and 31 require similar features. This is not taught or suggested by Dean. One skilled in the art would look upon the teachings of Dean merely as a way to bypass the controls on access to copy and save functions being provided at the client, not as a means to provide such controls. As such, the teachings of Dean are in direct contrast to the objectives of the present invention.

As noted by the Office Action, Dean states “In Netscape, Java applets can name only those functions and variables explicitly exported to the Java subsystem.” (See page 190, right-side column of Dean.) Dean does not say whether copy or save functions in particular are exported to the Java subsystem. However, in Netscape, copy and save functions are exported to the Java subsystem and would hence be available to Java applets running on a client unless, as in the present invention, some additional functionality is provided to suppress access to such functions which are otherwise available in respect to particular data sets.

Like Yourdon, Wobber’s system for authenticating requests for access to data is not relevant to the present invention as authority to access data is not in question. The present invention would generally operate after access to data has been granted by a server. That is, any teachings by Wobber would apply before the present invention

BRAMHILL et al.--Application No. 09/091,735

even comes into operation. In that respect, the teachings of Wobber would not add anything to the similarly irrelevant teachings of Yourdon and Dean.

Richardson is only discussed briefly with respect to dependent claims 5, 6 and 17. IMPRIMATUR is merely identified and is not discussed in any detail beyond its identification. Appellant respectfully submits that Richardson and IMPRIMATUR each fails to remedy any of the deficiencies of the combination of Yourdon, Dean and Wobber discussed above.

Appellant respectfully requests that the Examiner clarify which IMPRIMATUR reference is being used in the rejection and provide details regarding how this reference may relate to the present invention. Appellant notes, for example, that the document IMPRIMATUR: "Protection of Technological Measures" was published in November 1998 (i.e., after the March 18, 1998 international filing of the subject application). In any event, no IMPRIMATUR reference discloses or suggests running a program portion at a client to generate and upload a request for access to a data set; running of a program portion (separate from the data set being accessed) to unprotect the data set and to restrict copy or save functions at the client when the data set is otherwise unprotected at the client.

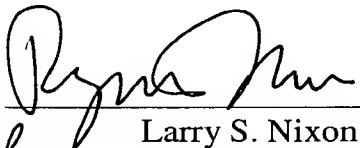

Accordingly, Appellant respectfully submits that claims 1-8, 12, 14-18, 21-26 and 28-31 are not obvious over Yourdon, Dean, Wobber, Richardson and IMPRIMATUR respectfully requests that the rejection of these claims under 35 U.S.C. §103 be reversed.

CONCLUSION

For all of the reasons set forth above, it is respectfully requested that this appeal be granted and that the rejections discussed above be reversed.

Respectfully submitted,

NIXON & VANDERHYE P.C.

By:  4/4/26
 Larry S. Nixon
Reg. No. 25,640

LSN/sje
1100 North Glebe Road, 8th Floor
Arlington, VA 22201-4714
Telephone: (703)816-4000
Facsimile: (703)816-4100

APPENDIX OF CLAIMS ON APPEAL

1. A method of protecting data sent from a server to a client, said method comprising:

running a program portion at the client, the program portion generating and uploading to the server a request for access to data;

cryptographically protecting the data;

sending the cryptographically protected data to the client; and

after the running of the program portion has begun and under control of the program portion at the client, converting the cryptographically protected data to an unprotected form and selectively controlling access to copy or save functions at the client in respect of the data in its unprotected form.

2. A method as in claim 1 wherein cryptographically protecting the data comprises protecting the data by encryption.

3. A method as in claim 1 wherein cryptographically protecting the data comprises protecting the integrity of the data cryptographically.

4. A method as in claim 3 wherein the integrity of the data is achieved by hashing.

5. A method as in claim 1 including authenticating that the client is permitted to receive the data.
6. A method as in claim 1 including identifying the client to the server before the data are sent to the client.
7. A method as in claim 1 including:
generating the program portion at a server,
downloading the program portion to the client, and
running the program portion on the client such that a request is uploaded to the server for a file containing the cryptographically protected data.
8. A method as in claim 7 wherein the program portion is generated in response to, and corresponds with, an earlier received request for access to the data.
12. A method as in claim 1 wherein the data are sent to the client from the server through a network.
14. A method as in claim 7 wherein the program portion includes data concerning a cryptographic key, and the method including using the key to render the downloaded cryptographically protected data into an unprotected form.

15. A method as in claim 1 wherein the server and the client each hold data corresponding to a cryptographic key and a machine identifier for uniquely identifying the client, the method including:

 sending a challenge to the client, such that it generates a signed response as a cryptographic function of the key and the machine identifier held therein,

 generating from the cryptographic key and machine identifier held associated with the server, a corresponding signed response as a cryptographic function of the key and the machine identifier,

 comparing the signed responses from the client and the server, and if they correspond, performing the cryptographic protection of the data with the key, and

 converting the cryptographically protected data into an unprotected form at the client with the key.

16. A method as in claim 1 wherein the data is steganographically marked.

17. A method as in claim 1 including registering the client with the server.

18. A method as in claim 1 including:

 determining a machine identifier of the client by analysing its hardware and/or its software configuration,

 transmitting the machine identifier to the server,

combining the transmitted machine identifier with a cryptographic key to form a unique determinator for the client,

transmitting the unique determinator to the client, to be stored therein for use subsequently in identifying the client to the server, to permit encrypted data to be downloaded thereto from the server.

21. A data storage medium storing copy protected data on the client received by a method according to claim 1.

22. A method of downloading encrypted data from a server to a client, said method including:

registering the client with the server by

determining a machine identifier of the client by analysing its hardware and/or its software configuration,

transmitting the machine identifier to the server,

combining the transmitted machine identifier with a cryptographic key to form a unique determinator for the client, and

transmitting the unique determinator to the client, to be stored therein for use subsequently in identifying the client to the server, to permit encrypted data to be downloaded thereto from the server,

subsequently identifying the client to the server on the basis of the unique determinator; and then

downloading data encrypted by means of the cryptographic key to the identified client, for decryption by the client using the key from the unique determinator.

23. A method as in claim 22 including decrypting the downloaded data at the client using the key from the unique determinator.

24. A method as in claim 22 wherein the client is identified to the server by:
again determining the machine identifier for the client,
comparing it with the machine identifier included in said unique determinator,
and
signalling to the server on the basis of the outcome of the comparison.

25. A method as in claim 22 including authenticating the client to the server prior to downloading of the encrypted data.

26. A method as in claim 25 including;
generating a challenge,
generating a response as a predetermined cryptographic function of the cryptographic key for the client as held by the server, and as a function of the key included in the unique determinator stored in the client, and
authenticating the client on the basis of the outcome of the comparison.

28. A server for providing access to data sets in a protected form, the server comprising:

an input for receiving a request for access to a data set;

protecting means for cryptographically protecting the requested data set; and

generating means for generating a program portion for sending to the source of the access request,

wherein said program portion is operable and after the program portion is permitted to run at the source of the access request, in use:

to generate a request for access to the cryptographically protected data set;

on receipt of the cryptographically protected data set, to convert it into an unprotected form; and

to selectively control access to copy or save functions in respect of the data set when in said unprotected form.

29. A computer program carrier medium containing a computer program which implements the functions of the server in claim 28 when installed and run on a server.

30. A method of protecting data downloaded from a server computer to a client computer, said method comprising:

downloading a protected copy of requested data from a server to a client; and running a program at the client so that after running the program at the client has begun at the client, the program serves to both: (a) unprotect the downloaded data thereby to provide access to an unprotected copy of the requested data, and (b) suppress client computer copy and save functions with respect to the unprotected copy of the requested data.

31. A method of protecting data sent from a server to a client, said method comprising:

running a program portion at the client, the program portion generating and uploading to the server a request for access to data;

cryptographically protecting the data;

sending the cryptographically protected data to the client; and

after access to the program portion is permitted and under control of the program portion, converting the cryptographically protected data to an unprotected form and restricting or preventing access to copy or save functions at the client in respect of the data in its unprotected form.

32. A server for providing access to data sets in a protected form, the server comprising:

an input for receiving a request for access to a data set;

protecting means for cryptographically protecting the requested data set; and

generating means for generating a program portion for sending to the source of the access request,

wherein after access to the program portion is permitted and said program portion is operable, in use:

to generate a request for access to the cryptographically protected data set;

on receipt of the cryptographically protected data set, to convert it into an unprotected form; and

to restrict or prevent access to copy or save functions in respect of the data set when in said unprotected form.

33. A method of protecting data downloaded from a server computer to a client computer, said method comprising:

downloading a protected copy of requested data from a server to a client; and

running a program at the client after access to the program is permitted to both:

(a) unprotect the downloaded data thereby to provide access to an unprotected copy of the requested data, and (b) restrict or prevent client computer copy and save functions with respect to the unprotected copy of the requested data.